Requested Patent: GB2292470A

Title: ROM PATCHING ;

Abstracted Patent: GB2292470 ;

Publication Date: 1996-02-21 ;

Inventor(s): LARRI GUY ;

Applicant(s): ADVANCED RISC MACH LTD (GB) ;

Application Number: GB19940016815 19940819 ;

Priority Number(s): GB19940016815 19940819 ;

IPC Classification: G06F11/20 ;

Equivalents: ;

ABSTRACT:

A data processing system having a central processor unit core 8 operating under control of program instructions stored within a read only memory 10 may be subject to mistakes within the program instructions. In order to deal with this, patch program instructions are stored within a patch memory 6. A breakpoint detector 14 monitors the address bus to the read only memory 10 and detects when an access is made to an address known to contain a mistake. Upon such an access, an interrupt handler 16 serves to divert processing to the patch stored within the patch memory 6 by writing a new value into a program counter register 12 coupled to the central processor unit core 8. At the end of the patch, control is returned to the program instructions of the read only memory 10.

GB 2292470A

(12) UK Patent Application (19) GB (11) 2 292 470 (13) A

(43) Date of A Publication 21.02.1996

(21) Application No 9416815.0

(22) Date of Filing 19.08.1994

(51) INT CL[6]
G06F 11/20

(52) UK CL (Edition O)
G4A AEF

(56) Documents Cited
GB 2250838 A GB 2245397 A GB 2122780 A
GB 1346219 A EP 0553733 A2 EP 0428005 A2

(58) Field of Search
UK CL (Edition M) G4A AEF
INT CL[5] G06F 11/20

(71) Applicant(s)
Advanced Risc Machines Limited

(Incorporated in the United Kingdom)

Fulbourn Road, Cherry Hinton, CAMBRIDGE, CB1 4JN, United Kingdom

(72) Inventor(s)
Guy Larri

(74) Agent and/or Address for Service
D Young & Co
21 New Fetter Lane, LONDON, EC4A 1DA, United Kingdom

(54) Rom patching

(57) A data processing system having a central processor unit core 8 operating under control of program instructions stored within a read only memory 10 may be subject to mistakes within the program instructions. In order to deal with this, patch program instructions are stored within a patch memory 6. A breakpoint detector 14 monitors the address bus to the read only memory 10 and detects when an access is made to an address known to contain a mistake. Upon such an access, an interrupt handler 16 serves to divert processing to the patch stored within the patch memory 6 by writing a new value into a program counter register 12 coupled to the central processor unit core 8. At the end of the patch, control is returned to the program instructions of the read only memory 10.

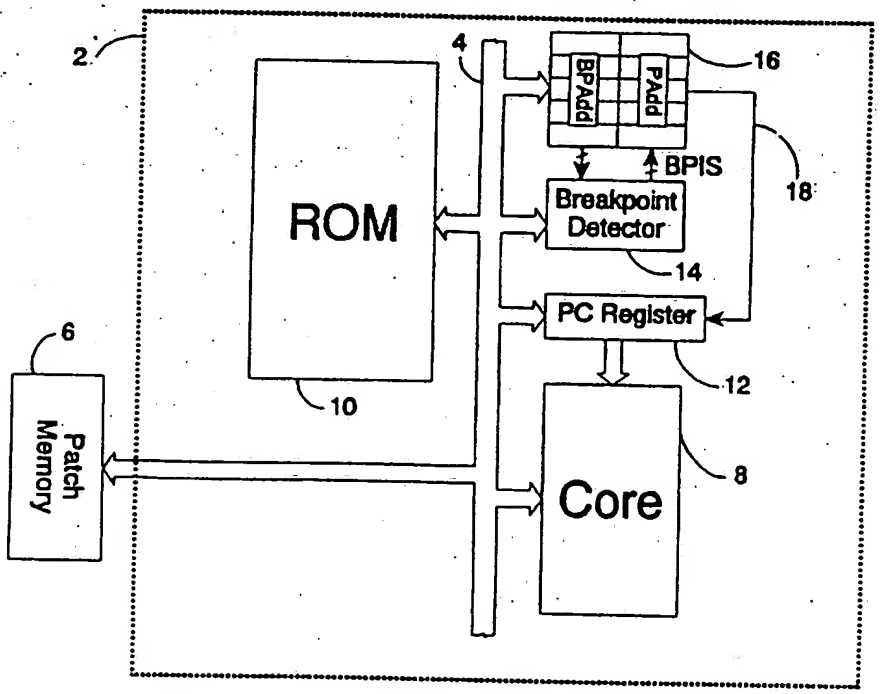
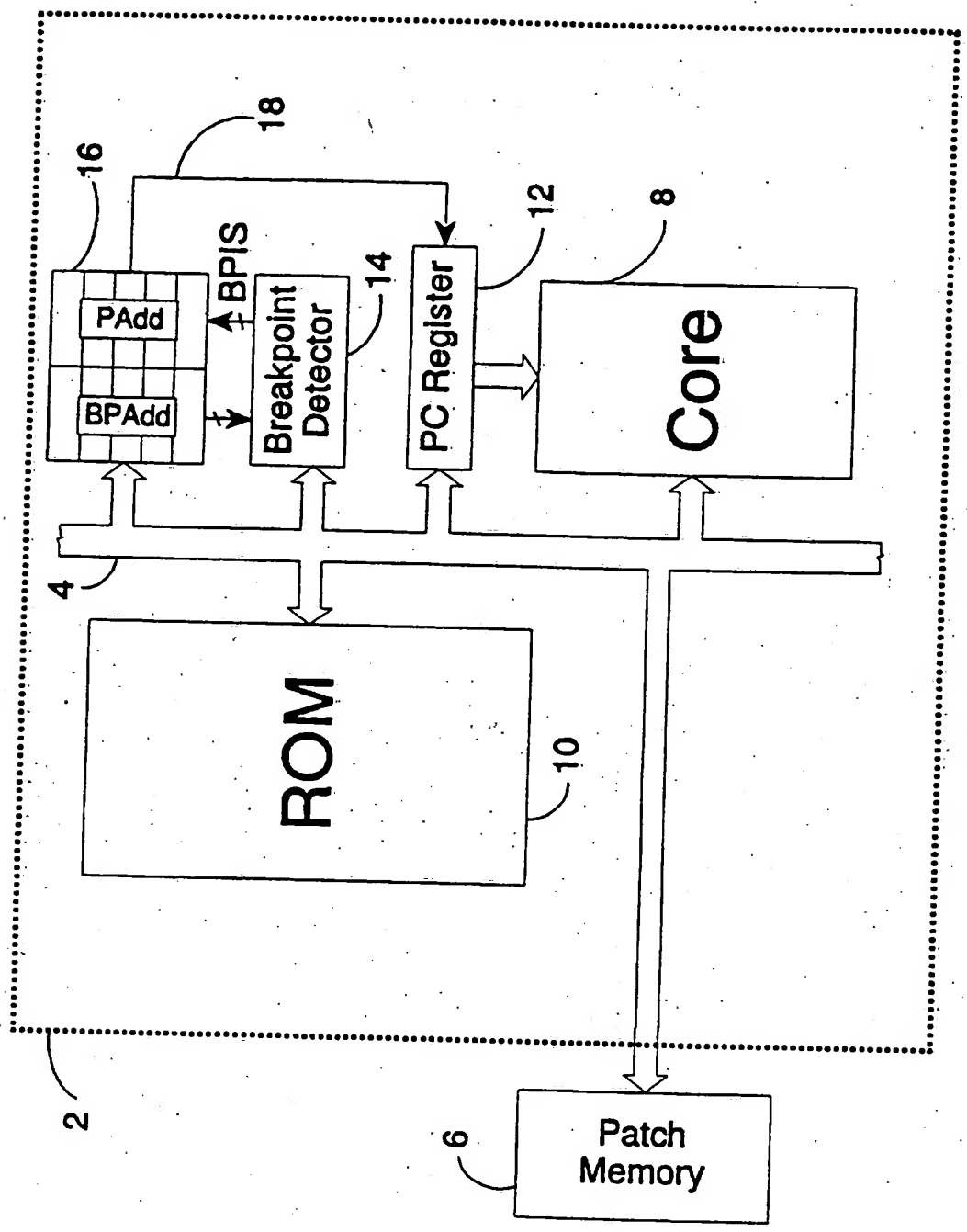


Fig.1

GB 2 292 470 A

Fig.1


18
16
12
8
BPIS
PAdd
BPAdd
Breakpoint Detector
14
PC Register
Core
4
ROM
10
2
6
Patch Memory

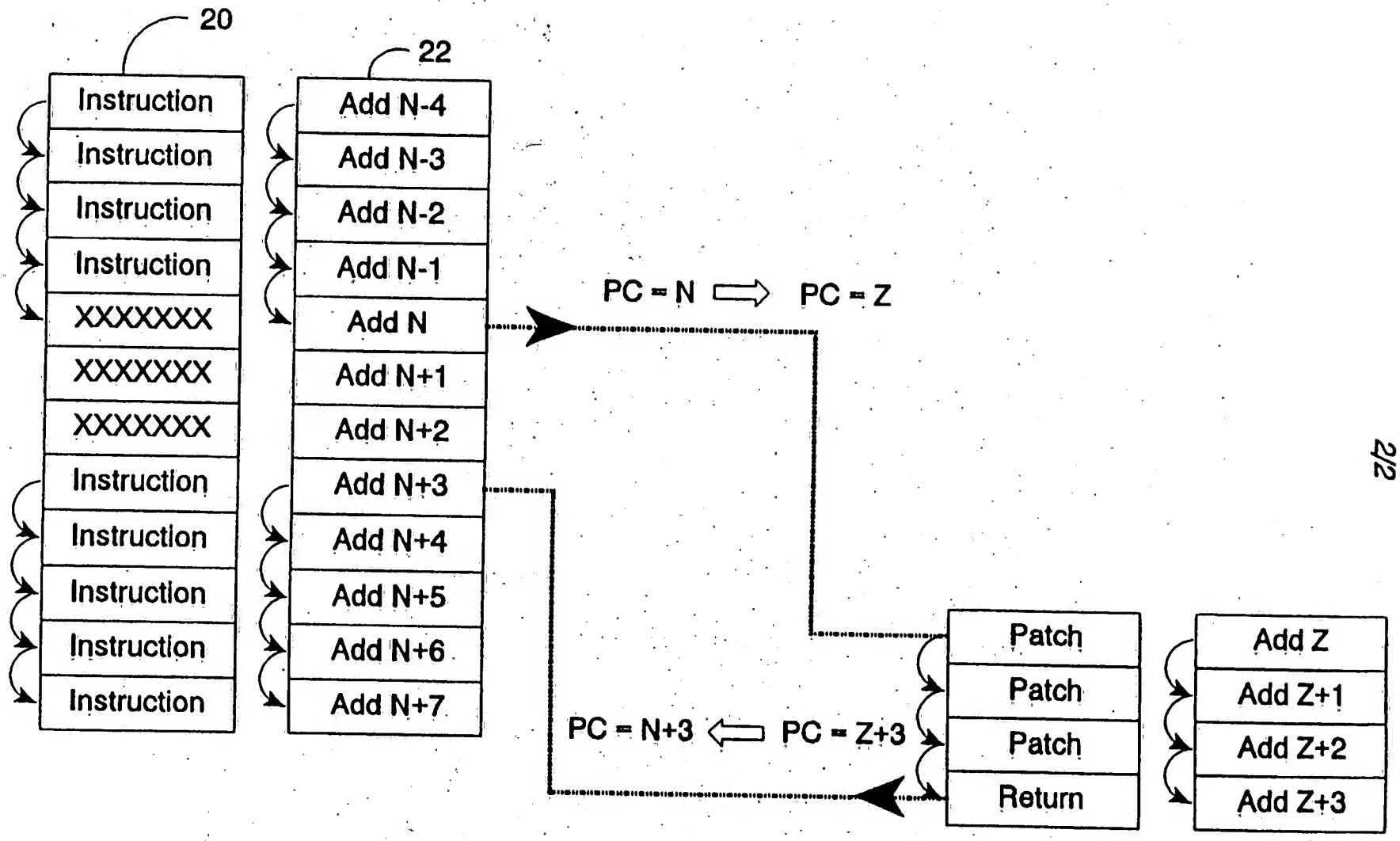

20
22
Instruction
Instruction
Instruction
Instruction
XXXXXXX
XXXXXXX
XXXXXXX
Instruction
Instruction
Instruction
Instruction
Instruction
Add N-4
Add N-3
Add N-2
Add N-1
Add N
Add N+1
Add N+2
Add N+3
Add N+4
Add N+5
Add N+6
Add N+7
PC = N ⇨ PC = Z
PC = N+3 ⇦ PC = Z+3
Patch
Patch
Patch
Return
Add Z
Add Z+1
Add Z+2
Add Z+3

Fig.2

ROM PATCHING

This invention relates to the field of data processing. More particularly, this invention relates to data processing systems having a central processor unit core controlled by program instructions stored within a read only memory (ROM).

It is known to provide data processing systems including a central processor unit core controlled by program instructions stored within a ROM. Such arrangements are becoming increasingly common and are often used within mass-market electronic devices incorporating embedded microprocessor controllers.

A constant problem with the use of program instructions to control central processor unit cores is the presence of mistakes ("bugs") within the program instructions. Whilst considerable efforts are taken to thoroughly test program instructions, some mistakes inevitably tend to be overlooked and are only discovered at a later date. In order to deal with such mistakes that are recognised at a late stage, it is known to correct such mistakes with so called "patches". If the mistake is small, then it may be possible to directly replace the erroneous program instructions within a sequence of corrected instructions. Alternatively, the mistake may be such that the required solution is to insert a sub-routine call into the original instruction flow, so that the new sub-routine can perform the necessary corrections to the operation.

Whilst such approaches are possible when the program instructions are stored in a manner in which they may be altered, this is not a possibility when the program instructions are stored in ROM. If a ROM contains mistakes, then it must be replaced, since there is no mechanism for altering the program instructions stored by the ROM. Whilst this is inconvenient in itself, the problem becomes even more severe when the ROM forms part of a larger, more expensive integrated circuit where several functional elements are formed as a single integrated circuit. The waste involved in replacing such sophisticated integrated circuits is even greater.

Whilst one approach might be to avoid the use of ROMs, this would cause its own problems. ROMs are an efficient, inexpensive and robust way of storing program instructions that would be disadvantageous to

give up.

Viewed from one aspect this invention provides apparatus for processing data comprising:

a central processor unit core controlled by program instructions, each program instruction having an associated instruction address;

a read only memory for storing program instructions;

a further memory for storing patch program instructions;

an address bus for transmitting instruction addresses from said central processor unit core to said read only memory;

a breakpoint detector coupled to said address bus for detecting on said address bus a predetermined instruction address within said read only memory and thereupon generating a breakpoint interrupt signal;

an interrupt handler responsive to said breakpoint interrupt signal for passing control of said central processor unit core to a sequence of patch program instructions stored within said further memory to effect a program patch over one or more program instructions stored in said read only memory and then returning control of said central processor unit core to a return program instruction stored within said read only memory following said one or more program instructions.

The invention provides a mechanism whereby the advantages of the use of a ROM may be achieved whilst the problems of program instruction mistakes may be readily overcome. The system is provided with a further memory, which need only be small, for handling any patch program instructions that later become necessary as mistakes within the program instructions of the ROM are discovered. When such mistakes are discovered, the breakpoint detector and the interrupt handler can be programmed to prevent the mistakes being executed and instead divert control to the patch program instructions within the further memory.

The return program instruction need not be special in any way, it is simply the next instruction to be executed after the patch. The return program instruction may typically be positioned after the bug to be fixed, but could also be at any point within the program stored within the ROM or at a point in another program not stored in the ROM.

It will be appreciated that whilst the invention is particularly suited to correcting mistakes within program instructions, it may also

be used to provide a degree of flexibility in modifying/updating the program instructions for reasons that do not stem from a coding mistake; in this case the patch program instructions are actually updated code.

Whilst the further memory could take many different forms, such as a further small read only memory or a set of registers, it is preferred that said further memory comprises a random access memory. Such random access memory is often already provided within such systems to provide other functions, such as working memory for dynamic data storage.

As previously discussed, it is possible with large scale integration techniques to incorporate more than one functional element upon a single integrated circuit. Accordingly, it would be possible to fabricate the further memory as part of the system containing the central processor unit core and ROM. However, it yields an advantageous degree of flexibility when said further memory comprises a discrete integrated circuit.

As previously mentioned, the invention is particularly suited to embodiments in which said central processor unit core, said read only memory, said breakpoint detector and said interrupt handler are formed within a single integrated circuit.

With regard to the interrupt handler, this may be implemented as a hardware circuit element. Alternatively, the interrupt handler could be implemented as a software routine executing on the central processor unit core, e.g. an interrupt handler routine may be started that diverts to the appropriate patch in dependence upon the value in the program counter register at the time of the interrupt, or such interrupt handler software could in another way determine which patch should be used in dependence upon the state of the system.

In order that the breakpoint detector and the interrupt handler can efficiently serve to divert processing control to the patch program instructions, in preferred embodiments said breakpoint detector stores said predetermined address.

In a complementary manner said interrupt handler stores an associated start address of said patch program instructions within said further memory.

Whilst the system could provide the capability for a single

patch, it is preferred that multiple patches can be supported by providing that said breakpoint detector and said interrupt handler store a plurality of said predetermined addresses and a plurality of said associated start addresses.

In preferred embodiments of the invention said central processor unit core comprises a program counter register storing an address of a current program instruction to control said central processor unit core and wherein said interrupt handler responds to said breakpoint interrupt signal by writing an associated start address of said patch program instructions into said program counter register.

The interrupt handler may, in accordance with the above, readily force a jump to the patch program instructions by writing a new program counter value.

At the end of a sequence of patch program instructions, it is preferred that the patch itself returns control to the main program instructions within the read only memory by writing a return address into the program counter register.

Viewed from another aspect this invention provides a method of processing data comprising the steps of:

storing program instructions within a read only memory;

controlling a central processor unit core by program instructions, each program instruction having an associated instruction address;

storing patch program instructions within a further memory;

transmitting instruction addresses from said central processor unit core to said read only memory via an address bus;

detecting on said address bus a predetermined instruction address within said read only memory and thereupon generating a breakpoint interrupt signal;

in response to said breakpoint interrupt signal, passing control of said central processor unit core to a sequence of patch program instructions stored within said further memory to effect a program patch over one or more program instructions stored in said read only memory and then returning control of said central processor unit core to a return program instruction stored within said read only memory following said on or more program instructions.

An embodiment of the invention will now be described, by way of

example only, with reference to the accompanying drawings in which:

Figure 1 illustrates a data processing system having a central processor unit core operating under control of program instructions stored within a ROM; and

Figure 2 illustrates the flow of control when a patch is utilised to skip a mistake within a main sequence of program instructions.

Figure 1 illustrates an integrated circuit 2 comprising a plurality of functional units. The integrated circuit 2 has a central bus 4 (including an address bus) via which signals are passed inside the integrated circuit 2 and to an external further memory in the form of a patch memory 6.

The integrated circuit 2 incorporates a central processor unit core 8 responsive to program instructions and a ROM 10 storing such program instructions. The central processor unit core 8 is controlled by whichever program instruction is stored at the address indicated by the current value held by a program counter register 12.

A breakpoint detector 14 is coupled to the bus 4 and continuously monitors the address bits of the currently accessed address within the ROM 10 to detect the occurrence of the predetermined breakpoint address. An interrupt handler 16 stores a plurality of such breakpoint addresses (BPAdd) and corresponding patch addresses (PAdd). When the break point detector 14 detects one of its stored breakpoint addresses, it issues a breakpoint interrupt signal (BPIS) to the interrupt handler 16 that triggers the patch address corresponding to the detected breakpoint address to be written into the program counter register 12 via a line 18. The values of the break point addresses (BPAdd) and patch addresses (PAdd) may be stored within a volatile memory and loaded into the interrupt handler 16 via the bus 4 upon initialisation. Another possibility is that the breakpoint addresses (BPAdd) and patch addresses (PAdd) could be serially loaded via a JTAG type serial interface to the integrated circuit.

In Figure 1 the breakpoint addresses (BPAdd) and patch addresses (PAdd) are shown as store within the interrupt handler. As an alternative the breakpoint addresses (BPAdd) could be stored in the breakpoint detector 14 from where they are readily available for high speed use on every cycle to compare with the address value being supplied to the ROM 10. In contrast, since the patch addresses (PAdd)

are only infrequently needed, they could be stored elsewhere (e.g. in th patch memory 6) from where the can be recovered when needed.

Figure 2 illustrates the operation of the circuit of Figure 1. A sequence of program instructions 20 are stored within the ROM 10. These program instructions control the operation of the central processor unit core 8 as it sequentially steps through these instructions. As this operation progresses, the value of the program counter value stored within the program counter register 12 increments as illustrated by the sequence of address values 22.

Within the program instructions stored in the read only memory 10, erroneous program instructions (XXXXXXX) are stored at three addresses (Add N, Add N+1 and Add N+2). As an attempt is made to control the central processor unit core 8 upon the basis of the first of these instructions at address N, the breakpoint detector 14 recognises this as one of its predetermined addresses and issues a breakpoint interrupt signal (BPIS). This breakpoint interrupt signal forces the interrupt handler to write a new value into the program counter register 12, in this case, the program counter value is changed from N to Z.

The address Z points to a location in the patch memory 6 which contains an appropriate correcting set of code for the mistake starting at address N. By writing the address Z into the program counter register 12, control is passed to the patch program instructions and these execute in sequence until the end of the patch. The last instructions in the patch at address Z+3 is a return instruction that serves to write a return address value into the program counter register 12 for returning control to the program instructions of the ROM 10. In this case, the program instructions return control to the instruction of address N+3, whereupon the sequence of execution of the program instructions of the ROM 10 is recommenced.

Whilst in the illustration of Figure 2 the patch contains the same number of program instructions as the error it fixes, this need not necessarily be the case, e.g. the patch may be a lengthy sub-routine that in some way upgrades the performance of the integrated circuit 2 rather than correcting an error in the program codes stored within the read only memory 10. In addition, the patch is illustrated as sequentially executing in Figure 2; this need not necessarily be the

case. The patch may contain jumps and sub routine calls just as any normal piece of code.

CLAIMS

1. Apparatus for processing data comprising:

a central processor unit core controlled by program instructions, each program instruction having an associated instruction address;

a read only memory for storing program instructions;

a further memory for storing patch program instructions;

an address bus for transmitting instruction addresses from said central processor unit core to said read only memory;

a breakpoint detector coupled to said address bus for detecting on said address bus a predetermined instruction address within said read only memory and thereupon generating a breakpoint interrupt signal;

an interrupt handler responsive to said breakpoint interrupt signal for passing control of said central processor unit core to a sequence of patch program instructions stored within said further memory to effect a program patch over one or more program instructions stored in said read only memory and then returning control of said central processor unit core to a return program instruction.

2. Apparatus as claimed in claim 1, wherein said return program instruction is stored within said read only memory following said one or more program instructions.

3. Apparatus as claimed in any one of claims 1 and 2, wherein said further memory comprises a random access memory.

4. Apparatus as claimed in any one of claims 1, 2 and 3, wherein said further memory comprises a discrete integrated circuit.

5. Apparatus as claimed in any one of the preceding claims, wherein said central processor unit core, said read only memory, said breakpoint detector and said interrupt handler are formed within a single integrated circuit.

6. Apparatus as claimed in any one of the preceding claims, wherein said breakpoint detector stores said predetermined address.

7. Apparatus as claimed in any ne of the preceding claims, wherein said interrupt handler stores an associated start address of said patch program instructions within said further memory.

8. Apparatus as claimed in claims 6 and 7, wherein said breakpoint detector and said interrupt handler store a plurality of said predetermined addresses and a plurality of said associated start addresses.

9. Apparatus as claimed in any one of the preceding claims, wherein said central processor unit core comprises a program counter register storing an address of a current program instruction to control said central processor unit core and wherein said interrupt handler responds to said breakpoint interrupt signal by writing an associated start address of said patch program instructions into said program counter register.

10. Apparatus as claimed in claim 9, wherein a last instruction within a sequence of patch program instructions writes into said program counter register a return address for said return program instruction stored within said read only memory.

11. A method of processing data comprising the steps of:

storing program instructions within a read only memory;

controlling a central processor unit core by program instructions, each program instruction having an associated instruction address;

storing patch program instructions within a further memory;

transmitting instruction addresses from said central processor unit core to said read only memory via an address bus;

detecting on said address bus a predetermined instruction address within said read only memory and thereupon generating a breakpoint interrupt signal;

in response to said breakpoint interrupt signal, passing control of said central processor unit core to a sequence of patch program instructions stored within said further memory to effect a program patch over one or more program instructions stored in said read only

memory and then returning control of said central processor unit core to a return program instruction.

12. Apparatus for processing data substantially as hereinbefore described with reference to the accompanying drawings.

13. A method of processing data substantially as hereinbefore described with reference to the accompanying drawings.

Patents Act 1977 Examiner's report to the Comptroller under Section 17 (The Search report)	Application number GB 9416815.0
Relevant Technical Fields (i) UK Cl (Ed.M) G4A AEF (ii) Int Cl (Ed.5) G06F 11/20	Search Examiner MR S J PROBERT Date of completion of Search 27 SEPTEMBER 1994
Databases (see below) (i) UK Patent Office collections of GB, EP, WO and US patent specifications. (ii)	Documents considered relevant following a search in respect of Claims :- 1-13

Categories of documents

X:	Document indicating lack of novelty or of inventive step.	P:	Document published on or after the declared priority date but before the filing date of the present application.
Y:	Document indicating lack of inventive step if combined with one or more other documents of the same category.	E:	Patent document published on or after, but with priority date earlier than, the filing date of the present application.
A:	Document indicating technological background and/or state of the art.	&:	Member of the same patent family; corresponding document.

Category	Identity of document and relevant passages		Relevant to claim(s)
X	GB 2250838 A	(HONDA GIKEN KOGYO KK) see abstract	1-13
X	GB 2245397 A	(SCHLUMBERGER) see whole document	1-13
X	GB 2122780 A	(SHARP KK) see abstract	1,11 at least
X	GB 1346219	(HONEYWELL BULL) see whole document	1,11 at least
X	EP 0553733 A2	(SONY CORP) see abstract	1-13
X	EP 0428005 A2	(TOSHIBA) see abstract	1-3

Databases: The UK Patent Office database comprises classified collections of GB, EP, WO and US patent specifications as outlined periodically in the Official Journal (Patents). The on-line databases considered for search are also listed periodically in the Official Journal (Patents).